Aula 6 - Sincronização

Processos em um sistema distribuído geralmente buscam atingir cooperação e para tanto utilizam mecanismos de sincronização para que esta cooperação seja realizada de maneira correta.
Veremos:

- Sincronização de relógio
- Sequenciamento de eventos
- Exclusão mútua
- *Deadlock*
- Algoritmos de eleição

Sincronização de relógio

Cada computador possui seu próprio relógio. Como em um sistema distribuído os processos podem estar sendo executados em diversos computadores, os relógios destes computadores devem estar sincronizados. (Exemplo. Sistema de reservas online).
[Liskov1993] "Pratical Uses of Syncrhonized Clocks in Distributed Systems". Distributed Computing, Vol 6, pp. 211-219 (1993).

Como os relógios funcionam

Três componentes: cristal de quartzo que oscila em uma freqüência fixa, um registrador contador, um registrador constante (armazena uma constante baseada na freqüência de oscilação do cristal) Cada vez que o registrador contador é zero uma interrupção é causada.

Precisão dos relógios

Um relógio sempre trabalha de maneira constante, pois o cristal oscila em uma freqüência fixa. Entretanto, cristais diferentes podem oscilar diferentemente. A diferença pode ser extremamente pequena, mas em um certo período isto pode causar problemas. O relógio do computador pode assim se afastar de um relógio real (cálculo atravé das transições que o Césio 133 faz em um segundo foi definido em 1958 como sendo 9.192.631.770 transições do Csio 133. Foi utilizado este cálculo pois a terra tem um perí́odo de rotação não estável). Até 1995 houve uma diferença entre o relógio calculado através do período de rotação da terra com o do Césio 133 de 3mseg a cada 86.400 segundos.)

Relógios baseados em cristal tem uma diferença de 1 seg a cada 1000000 de segundos ou seja a cada 11,6 dias. Desta forma o relógio do computador deve ser resincronisado de tempos em tempos.
Para ser mais preciso suponha que a hora real seja t, e o tempo de um relógio p é $C_p(t)$. Se todos os relógios no mundo fossem perfeitamente sincronizados então $C_p(t) = t$ para todos p e todos t. Isto é se C denota o valor de um relógio, idealmente teriamos dC/dt=1. Como este tipo de situação não é real podemos dizer que um relógio está correto se existir um valor ρ que representa o máximo que o relógio desvia de dC/dt, ou seja: (1-ρ)<=(dC/dt)<=(1+ρ).
Relógio muito rápido, relógio muito lento, relógio perfeito. (figura)
Como um sistema distribuído possui diversos relógios, que não estão sincronizados, deve-se sincronizá-los periodicamente. Considere um intervalo de tempo Dt, e dois relógios, um muito rápido e um muito devagar, a diferenca máxima entre estes dois relógios deve atingir 2ρDt. Desta forma, se quisermos que dois relógios não possuam diferença de tempo maior que σ os relógios devem ser sincronizados periodicamente, e este período deve ser $\sigma/2\rho$.
Dois tipos de sincronização de relógios:

- Sincronização do relógio do computador com relógios de tempo real (ou externos), usando UTC (*Universal Coordinated Time*).
- Sincronização de relógios de nodos diferentes em um sistema (internos).

Considerações

- Relógios estão sincronizados se a diferença entre os dois não for maior que uma constante definida σ.
- Computadores devem conhecer o valor dos relógios dos outros computadores no sistema.
- Ao ler o valor, problemas podem acontecer durante a comunicação deste valor.
- Importante: relógios não podem nunca voltar no tempo. Diminui a velocidade do relógio gradualmente, ou aumenta a velocidade. Ex. se a interrupção adicionasse 8 mseg, então poderia adicionar 7 ou 9.

Algoritmos de sincronização

Podem ser centralizados ou distribuídos. Em algoritmos centralizados existe a figura de um servidor que mantêm o valor real do relógio e o mesmo informa os demais computadores a respeito deste valor.

Centralizados

Duas formas:

- Servidor passivo: espera ser perguntado "hora=?". Então responde "hora=T", onde T é a hora no servidor. Assuma que o cliente enviou a mensagem perguntando pela hora no tempo T0 e recebeu a mensagem contendo a hora no servidor no tempo T1 (T0 e T1 são tempos do relógio do cliente). Então a melhor estimativa do novo horário para o cliente é T+(T1-T0)/2. Uma forma de melhorar esta estimativa é incluir o tempo que o servidor leva para tratar a mensagem "hora=?" enviada pelo cliente, suponha que este tempo seja I. Desta forma o tempo no cliente poderia ser ajustado para T+(T1-T0-I)/2.

  Cristian[1989] propôs uma variação deste método para incluir diversos cálculos (T1-T0), sendo que alguns destes valores que ultrapassam determinado limite são desconsiderados. Usa-se a média destes valores e esta média dividida por dois é o valor a ser adicionada a T. Pode-se ainda utilizar o menor valor de (T1-T0) como valor a ser considerado.

- Servidor ativo: *broadcasts* o valor da hora para todos os computadores. Neste algoritmo, considere que o servidor sabe o tempo aproximado de transmissão entre ele e cada um dos clientes (Ta), então de tempo em tempo o servidor *broadcasts* a mensagem "hora=T+Ta" para cada um dos clientes. O problema neste método é que se a mensagem leva muito mais tempo do que Ta para chegar ao cliente, o cliente vai atualizar seu relógio para um valor incorreto.

  O algoritmo utilizado no *Berkeley Unix*, resolve este problema da seguinte forma. De tempo em tempo o servidor envia uma mensagem "hora=?" a todos os clientes. O servidor tem conhecimento prévio de qual o tempo que uma mensagem leva para chegar do cliente a ele. A partir dai o servidor calcula a média das horas em todos os clientes (inclusive a sua). Horas que estão fora de certos limites são desconsideradas para não causarem problemas no cálculo da média. O valor calculado é o valor que todos os relógios deveriam ser ajustados. O servidor ajusta o seu relógio e envia uma mensagem para cada um dos clientes informando em quanto eles devem ajustar seus relógios.

Problemas com este tipo de algoritmo: *single-point failure*, "escalabilidade".

Distribuídos

Uma solução para sincronizar os relógios dos computadores seria utilizar o valor da hora real e cada um dos computadores seria ajustado usando este valor. Só que usar relógios verdadeiros pode trazer ainda mais imprecisão.

Dois algoritmos distribuídos para ajustar os relógios:

- Algoritmo de cálculo da media global: neste algoritmo cada processo que mantém o relógio do computador *broadcasts* uma mensagem com o valor de seu relógio em um tempo T0+iR, para todos os outros processos no sistema. T0 é um valor combinado no passado, i representa os intervalos que os processos irão sincronizar seus relógios, e R representa de quanto em quanto tempo a sincronização vai acontecer (leva em conta o número de processos no sistema e o máximo que se deseja de diferença entre os relógios).

  Depois de fazer o *broadcast*, o processo espera um intervalo T para receber as mensagens dos outros processos. Após o intervalo T, o processo então recalcula o seu relógio. Para melhorar a precisão do cálculo, alguns valores podem ser desconsiderados, e.g. maiores que um limite, ou os m maiores e os m menores.

- Algoritmo de cálculo de média local: o algoritmo anterior tem o problema de necessitar um sistema de *broadcast* de mensagens. Desta forma ele é bom para pequenas redes. Neste algoritmo cada processo troca informações com os processos vizinhos e calcula seu novo horário a partir destes relógios.